John Høeg Semester prøve

# MainApp

**package** gui;

**import** controller.Controller;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.TabPane;

**import** javafx.scene.control.TabPane.TabClosingPolicy;

**import** javafx.scene.layout.BorderPane;

**import** javafx.stage.Stage;

**import** java.util.ArrayList;

**import** java.util.Optional;

**import** javafx.beans.value.ChangeListener;

**import** javafx.collections.ListChangeListener;

**import** javafx.collections.ObservableList;

**import** javafx.geometry.Insets;

**import** javafx.geometry.Pos;

**import** javafx.geometry.VPos;

**import** javafx.scene.control.Alert;

**import** javafx.scene.control.Alert.AlertType;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.ButtonType;

**import** javafx.scene.control.CheckBox;

**import** javafx.scene.control.ChoiceBox;

**import** javafx.scene.control.ComboBox;

**import** javafx.scene.control.Label;

**import** javafx.scene.control.ListView;

**import** javafx.scene.control.SelectionMode;

**import** javafx.scene.control.TextArea;

**import** javafx.scene.control.TextField;

**import** javafx.scene.layout.GridPane;

**import** javafx.scene.layout.HBox;

**import** java.time.LocalDate;

**import** java.time.LocalDateTime;

**import** java.time.format.DateTimeFormatter;

**import** java.time.temporal.ChronoUnit;

**import** java.util.ArrayList;

**import** application.Invalideplads;

**import** application.Parkeringshus;

**import** application.Parkeringsplads;

**import** application.Bil;

**import** application.EnumBilmærke;

**import** storage.Storage;

**public** **class** MainApp **extends** Application {

**public** **static** **void** main(String[] args) {

Application.*launch*(args);

}

@Override

**public** **void** init() {

Controller.*initStorage*();

}

@Override

**public** **void** start(Stage stage) {

stage.setTitle("Yatzy");

GridPane pane = **new** GridPane();

**this**.initContent(pane);

Scene scene = **new** Scene(pane);

stage.setScene(scene);

stage.setResizable(**false**);

stage.show();

}

// -------------------------------------------------------------------------

// Attributes

**private** Label lblparkeringshuse, lbloptagnepladser, lblbilmednummer;

**private** ListView<Parkeringshus> lvwParkeringshuse;

**private** ListView<Bil> lvwOptagnePladser;

**private** TextField txtOpretBil;

// -------------------------------------------------------------------------

**private** **void** initContent(GridPane pane) {

pane.setGridLinesVisible(**false**);

pane.setPadding(**new** Insets(10));

pane.setHgap(10);

pane.setVgap(10);

pane.setStyle("-fx-border-color: black");

lblparkeringshuse = **new** Label("Parkeringshuse");

pane.add(lblparkeringshuse, 0, 0);

lbloptagnepladser = **new** Label("Optagne pladser");

pane.add(lbloptagnepladser, 4, 0);

lvwParkeringshuse = **new** ListView<>();

pane.add(lvwParkeringshuse, 0, 1);

lvwParkeringshuse.setPrefWidth(200);

lvwParkeringshuse.setPrefHeight(200);

lvwParkeringshuse.getItems().addAll(Controller.*getParkeringshus*());

lvwOptagnePladser = **new** ListView<>();

pane.add(lvwOptagnePladser, 4, 1);

lvwOptagnePladser.setPrefWidth(200);

lvwOptagnePladser.setPrefHeight(200);

lvwOptagnePladser.getItems().addAll(Controller.*getBil*());

lblbilmednummer = **new** Label("Opret bil med nummer");

pane.add(lblbilmednummer, 0, 2);

}

}

# Controller

**package** controller;

**import** java.util.ArrayList;

**import** application.Invalideplads;

**import** application.Parkeringshus;

**import** application.Parkeringsplads;

**import** application.Bil;

**import** application.EnumBilmærke;

**import** storage.Storage;

**public** **class** Controller {

// Create metoder

**public** **static** Parkeringsplads createParkeringsplads(**int** nummer) {

Parkeringsplads parkeringsplads = **new** Parkeringsplads(nummer);

**return** parkeringsplads;

}

**public** **static** Invalideplads createInvalideplads(**int** nummer, **double** areal) {

Invalideplads invalideplads = **new** Invalideplads(nummer, areal);

**return** invalideplads;

}

**public** **static** Parkeringshus createParkeringshus(String addresse) {

Parkeringshus parkeringshus = **new** Parkeringshus(addresse);

Storage.*addParkeringshuse*(parkeringshus);

**return** parkeringshus;

}

**public** **static** Bil createBil(String regNr) {

Bil bil = **new** Bil(regNr, **null**);

Storage.*addBiler*(bil);

**return** bil;

}

// Delete Metoder

**public** **static** **void** deleteParkeringsplads(Parkeringsplads parkeringsplads) {

parkeringsplads.getParkeringshus().removeParkeringspladser(parkeringsplads);

}

**public** **static** **void** deleteInvalideplads(Invalideplads invalideplads) {

invalideplads.getParkeringshus().removeParkeringspladser(invalideplads);

}

// public static void deleteParkeringshus(Parkeringshus parkeringshus) {

//

// }

// public static void deleteBil(Parkeringshus parkeringshus) {

//

// }

// Update Metoder

**public** **static** **void** updateParkeringsplads(Parkeringsplads parkeringsplads, **int** nummer) {

parkeringsplads.setNummer(nummer);

}

**public** **static** **void** updateInvalideplads(Invalideplads invalideplads, **int** nummer, **double** areal) {

invalideplads.setAreal(areal);

invalideplads.setNummer(nummer);

}

**public** **static** **void** updateParkeringshus(Parkeringshus parkeringshus, String addresse) {

parkeringshus.setAddresse(addresse);

}

**public** **static** **void** updateBil(Bil bil, String regNr) {

bil.setRegNr(regNr);

}

// Get Metoder

**public** **static** ArrayList<Parkeringshus> getParkeringshus() {

**return** Storage.*getParkeringshuse*();

}

**public** **static** ArrayList<Bil> getBil() {

**return** Storage.*getBiler*();

}

**public** **static** **void** initStorage() {

// Added Biler

Bil b1 = **new** Bil("AB 11 222", EnumBilmærke.***HONDA***);

Bil b2 = **new** Bil("EF 33 444", EnumBilmærke.***HONDA***);

Bil b3 = **new** Bil("BD 55 666", EnumBilmærke.***AUDI***);

// Added Parkeringshus

Parkeringshus p1 = **new** Parkeringshus("Havnegade 12, 8000 Aarhus");

// Added Parkeringsplads

**for** (**int** i = 1; i <= 100; i++) {

Parkeringsplads pp = **new** Parkeringsplads(i);

p1.addParkeringspladser(pp);

}

// Added Invalideplads

**for** (**int** i = 101; i <= 110; i++) {

Invalideplads ip = **new** Invalideplads(i, 18);

p1.addParkeringspladser(ip);

}

// Tilføj Biler til parkeringshus

p1.createParkeringsplads(3).addBil(b1);

p1.createParkeringsplads(78).addBil(b2);

p1.createParkeringsplads(105).addBil(b3);

}

}

# Storage

**package** storage;

**import** java.util.ArrayList;

**import** application.Invalideplads;

**import** application.Parkeringshus;

**import** application.Parkeringsplads;

**import** application.Bil;

**public** **class** Storage {

**private** **static** ArrayList<Parkeringshus> *parkeringshuse* = **new** ArrayList<>();

**private** **static** ArrayList<Bil> *biler* = **new** ArrayList<>();

**public** **static** ArrayList<Parkeringshus> getParkeringshuse() {

**return** **new** ArrayList<Parkeringshus>(*parkeringshuse*);

}

**public** **static** **void** addParkeringshuse(Parkeringshus parkeringshus) {

*parkeringshuse*.add(parkeringshus);

}

**public** **static** ArrayList<Bil> getBiler() {

**return** **new** ArrayList<Bil>(*biler*);

}

**public** **static** **void** addBiler(Bil bil) {

*biler*.add(bil);

}

}

# Bil

**package** application;

**public** **class** Bil {

**private** String regNr;

**private** EnumBilmærke bilmærke;

**public** Bil(String regNr, EnumBilmærke bilmærke) {

**this**.regNr = regNr;

**this**.bilmærke = bilmærke;

}

@Override

**public** String toString() {

**return** regNr;

}

**public** String getRegNr() {

**return** regNr;

}

**public** **void** setRegNr(String regNr) {

**this**.regNr = regNr;

}

**public** EnumBilmærke getBilmærke() {

**return** bilmærke;

}

**public** **void** setBilmærke(EnumBilmærke bilmærke) {

**this**.bilmærke = bilmærke;

}

}

# Invalideplads

**package** application;

**import** java.time.LocalTime;

**import** application.Parkeringsplads;

**public** **class** Invalideplads **extends** Parkeringsplads{

**private** **double** areal;

**public** Invalideplads(**int** nummer, **double** areal) {

**super**(nummer);

**this**.areal = areal;

}

**public** **double** getAreal() {

**return** areal;

}

**public** **void** setAreal(**double** areal) {

**this**.areal = areal;

}

**public** **double** parkeringsPris(LocalTime slutTidspunkt) {

**double** pris = 0;

pris = **super**.parkeringsPris(slutTidspunkt) - **super**.parkeringsPris(slutTidspunkt);

**return** pris;

}

}

# Parkeringshus

**package** application;

**import** java.time.LocalTime;

**import** java.util.ArrayList;

**import** application.Parkeringsplads;

**public** **class** Parkeringshus {

**private** String addresse;

**private** **final** ArrayList<Parkeringsplads> parkeringspladser = **new** ArrayList<>();

**private** **double** saldo;

**private** ArrayList<String> optagetPladser = **new** ArrayList<>();

**public** Parkeringshus(String addresse) {

**this**.addresse = addresse;

}

**public** ArrayList<Parkeringsplads> getParkeringsplads() {

**return** **new** ArrayList<>(parkeringspladser);

}

**public** Parkeringsplads createParkeringsplads(**int** nummer) {

Parkeringsplads parkeringsplads = **new** Parkeringsplads(nummer);

**this**.addParkeringspladser(parkeringsplads);

**return** parkeringsplads;

}

**public** **void** addParkeringspladser(Parkeringsplads parkeringsplads) {

**if** (!parkeringspladser.contains(parkeringsplads)) {

parkeringspladser.add(parkeringsplads);

parkeringsplads.setParkeringshus(**this**);

}

}

**public** **void** removeParkeringspladser(Parkeringsplads parkeringsplads) {

**if** (parkeringspladser.contains(parkeringsplads)) {

parkeringspladser.remove(parkeringsplads);

parkeringsplads.setParkeringshus(**null**);

}

}

**public** String getAddresse() {

**return** addresse;

}

**public** **void** setAddresse(String addresse) {

**this**.addresse = addresse;

}

**public** **int** antalLedigePladser() {

**int** antalPladser = 0;

**return** antalPladser;

}

**public** **int** findPladsMedBil(String regNummer) {

**boolean** found = **false**;

**int** i = 0;

**while** (!found && i < parkeringspladser.size()) {

**if** (parkeringspladser.get(i).getBil().equals(regNummer)) {

found = **true**;

} **else** {

i++;

}

}

**if** (found) {

**return** parkeringspladser.get(i).getNummer();

} **else** {

**return** -1;

}

}

**public** **double** getPrisForParkering(ArrayList<Parkeringsplads> parkering, LocalTime slutTidspunkt) {

**double** pris = 0;

**for** (Parkeringsplads park : parkering) {

pris = park.parkeringsPris(slutTidspunkt);

}

**return** pris;

}

**public** **double** getSaldo() {

**return** saldo;

}

**public** **void** setSaldo(**double** saldo) {

**this**.saldo += saldo;

}

**public** **int** findAntalBiler(EnumBilmærke bilmærke) {

**int** antal = 0;

**for** (**int** i = 0; i < getParkeringsplads().size(); i++) {

**if** (getParkeringsplads().get(i).equals(bilmærke)) {

antal++;

}

}

**return** antal;

}

// Kunne jeg ikke huske hvordan man gjorde

// public String optagnepladser(ArrayList<String> optagetPladser) {

//

//

// return addresse ;

// }

}

# Parkeringsplads

**package** application;

**import** java.time.LocalTime;

**import** java.time.temporal.ChronoUnit;

**import** java.util.ArrayList;

**import** application.Bil;

**import** application.Parkeringshus;

**import** application.Invalideplads;

**public** **class** Parkeringsplads {

**private** **int** nummer;

**private** LocalTime ankomst;

**private** **final** ArrayList<Bil> biler = **new** ArrayList<>();

**private** Parkeringshus parkeringshus;

**private** Invalideplads invalideplads;

**public** Parkeringsplads(**int** nummer) {

**this**.nummer = nummer;

**this**.ankomst = **null**;

}

**public** Parkeringshus getParkeringshus() {

**return** parkeringshus;

}

**public** Invalideplads getInvalideplads() {

**return** invalideplads;

}

**public** **int** getNummer() {

**return** nummer;

}

**public** **void** setNummer(**int** nummer) {

**this**.nummer = nummer;

}

**public** LocalTime getAnkomst() {

**return** ankomst;

}

**public** **void** setAnkomst(LocalTime ankomst) {

**this**.ankomst = ankomst;

}

**public** ArrayList<Bil> getBil() {

**return** **new** ArrayList<>(biler);

}

**public** Bil createBil(String regNr, LocalTime ankomst, EnumBilmærke bilmærke) {

Bil bil = **new** Bil(regNr, bilmærke);

**this**.ankomst = ankomst;

**this**.addBil(bil);

**return** bil;

}

**public** **void** addBil(Bil bil) {

**if** (!biler.contains(bil)) {

biler.add(bil);

}

}

**public** **void** removeBil(Bil bil) {

**if** (biler.contains(bil)) {

biler.remove(bil);

}

}

**public** **void** setInvalideplads(Invalideplads invalideplads) {

**this**.invalideplads = invalideplads;

}

**public** **void** setParkeringshus(Parkeringshus parkeringshus) {

**if** (**this**.parkeringshus != parkeringshus) {

Parkeringshus oldParkeringshus = **this**.parkeringshus;

**if** (oldParkeringshus != **null**) {

oldParkeringshus.removeParkeringspladser(**this**);

}

**this**.parkeringshus = parkeringshus;

**if** (parkeringshus != **null**) {

parkeringshus.addParkeringspladser(**this**);

}

}

}

**public** **double** parkeringsPris(LocalTime slutTidspunkt) {

**double** pris = 0;

**double** minutesBetween = ChronoUnit.***MINUTES***.between(ankomst, slutTidspunkt);

pris = minutesBetween % 10 \* 6;

**return** pris;

}

**public** **void** hentBil(LocalTime slutTidspunkt, Bil bil) {

parkeringshus.setSaldo(parkeringsPris(slutTidspunkt));

removeBil(bil);

}

}

# EnumBilMærke

**package** application;

**public** **enum** EnumBilmærke {

***HONDA***, ***TOYOTA***, ***SKODA***, ***NISSAN***, ***AUDI***

}